Design of Multiplierless Decimation Filters Using an Extended Search of Cyclotomic Polynomials

M. Laddomada, D. E. Troncoso, and G. Jovanovic Dolecek

***Abstract*—This brief addresses the design of multiplierless decimation filters using an extended set of cyclotomic polynomials (CPs) as constituent filters. It extends the results presented in a companion paper by one of the coauthors to CPs with indexes in the set {1, . . . , 200}, and it presents the z-transfer functions of all CPs with indexes from 61 to 200. One of the key observations stemming from the results of this brief is that CPs with indexes in the set {105, . . . , 200} still have very effective coefficients, i.e., integers belonging to $\{-1, 0, +1\}$, but z-transfer functions have to be recursive. Regardless of the application to multirate filters considered in this brief, these polynomials can also be used for designing classical finite-impulse response filters. Moreover, this brief provides guidelines to simplify the design of constituent decimation filters in multistage architectures and to reduce computational complexity of the proposed filters. Finally, comparisons are given with respect to other techniques in the literature.**

***Index Terms*—Analog-to-digital (A/D) converter, cascaded integrator–comb (CIC) filter, comb, cyclotomic, cyclotomic polynomial (CP), decimation, decimation filter, multistage, polynomial, sigma–delta, sinc filters.**

I. Introduction and Objectives

The DESIGN of multistage decimation filters has recently received renewed interest spurred by the need of computationally efficient architectures for wide-band, multistandard, and reconfigurable receivers [1], [2]. Multistage decimation filters are also employed for decimating signals oversampled by noise-shaping sigma–delta analog-to-digital (A/D) converters [3].

Let us briefly present the architecture where these kind of filters find application by referring to Fig. 1. In Fig. 1(a), a baseband analog input signal $x(t)$ with bandwidth $[-B_x, +B_x]$ is oversampled by an A/D converter, whereas a digital signal is identified by $x(nT_o)$. Discrete time T_o can be easily found by relating sampling frequency f_o to signal bandwidth B_x as $f_o = (1/T_o) = 2\rho B_x$, where $\rho \geq 1$ is the oversampling ratio (notice that $\rho > 1$ for oversampled signals, whereas $\rho = 1$ for A/D converters operating at the Nyquist frequency). The normalized maximum frequency contained in the input signal is $f_c = (B_x/f_o) = (1/2\rho)$. With this setup, the sampled signal $x(nT_o)$ at the input of the first decimation filter $H(z)$ shows frequency components in the range $[-f_c, f_c]$, as pictorially

Manuscript received August 24, 2010; revised October 26, 2010; accepted November 26, 2010. Date of publication February 14, 2011; date of current version February 24, 2011. This paper was recommended by Associate Editor Y. Yu.

M. Laddomada is with the Electrical Engineering Department, Texas A&M University–Texarkana, Texarkana, TX 75505 USA (e-mail: mladdomada@tamut.edu).

D. E. Troncoso and G. Jovanovic Dolecek are with the Electronics Department, National Institute of Astrophysics, Optics and Electronics (INAOE) Puebla 72840, Mexico, (e-mail: dtroncoso@inaoep.mx; gordana@inaoep.mx).

Digital Object Identifier 10.1109/TCSII.2010.2104012

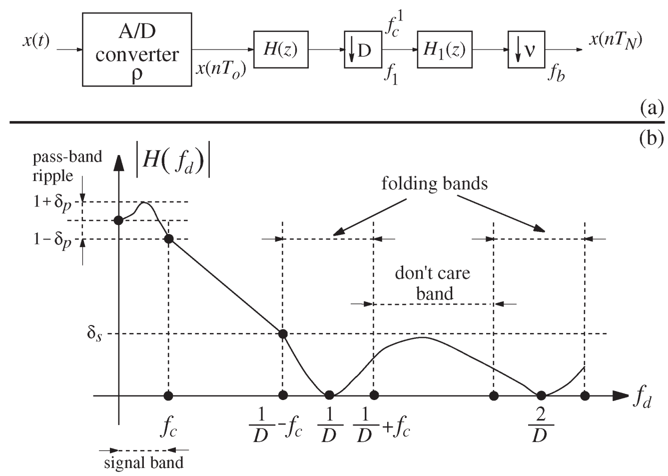

Fig. 1. (a) Architecture of the two-stage decimation structure. (b) Pictorial representation of the frequency response of the first decimation filter $H(z)$ along with the key frequency intervals to carefully consider in the design of the first decimation stage.

depicted in Fig. 1(b). The oversampling factor ρ is split among two decimation stages in such a way that $\rho = D \cdot \nu$ (D and ν are, respectively, the decimation factors of the first and second decimation stages).

Upon considering the pictorial frequency response in Fig. 1(b), it is worth noticing that, unlike classical filter design, the design of decimation filters calls for stringent constraints in folding bands defined as

$$\left[\frac{k}{D} - f_c,\, \frac{k}{D} + f_c\right],\ k = 1, \ldots, k_M = \left\lfloor \frac{D}{2} \right\rfloor \tag{1}$$

where f_c is the normalized signal bandwidth at the input of the first decimation filter. On the other hand, frequency intervals identified as do-not-care bands in Fig. 1(b) do not require stringent selectivity since any spurious signal falling within these bands will be rejected by subsequent filters in multistage decimation architectures.

The two basic design specifications for the first decimation filter $H(z)$ are a passband ripple δ_p defined (in decibels) as

$$R_p = -20\,\log_{10}\left(\frac{1-\delta_p}{1+\delta_p}\right) > 0 \tag{2}$$

and selectivity (in decibels) defined as

$$A_s = 20\,\log_{10}\left(\frac{\delta_s}{1+\delta_p}\right) \approx 20\,\log_{10}(\delta_s) \ll 0. \tag{3}$$

With this background, let us provide a quick survey of the recent literature. Excellent tutorials on the design of multirate filters can be found in [4] and [5], whereas an essential book on this topic is [6]. The design of classical FIR filters using cyclotomic polynomial (CP) prefilters was addressed in [7], whereas effective algorithms for the design of low-complexity classical FIR filters embedding CP prefilters were proposed in [8]–[10]. In [11], the authors proposed a new algorithm for a multiple constant multiplication problem producing solutions requiring up to 20% less addition and subtraction than the best known algorithms. An exact common subexpression elimination algorithm for optimum sharing of partial terms in multiple constant multiplication was proposed in [12].

The design of decimation filters based on CPs was proposed in a companion paper [13] using the set of the first 104 CPs. Compared with [13], the aim of this brief is as follows. First, we present z-transfer functions of CPs with indexes from 105 to 200, as well as z-transfer functions of CPs with indexes from 61 to 104, that were only referred to an internal report in the companion work [13]. We notice that these CPs do have coefficients in the set $\{-1, 0, +1\}$ when we allow recursive z-transfer functions. This property makes this polynomials very well suited to computationally efficient filter architectures. Moreover, in this brief, we extend the design procedure presented in the companion paper [13] by considering all the CPs with indexes in the set $\{1, \ldots, 200\}$ and present the design results in Section III. We also present some hints to reduce computational complexity of the designed decimation filters. Finally, Section IV draws the conclusion.

II. Optimization Algorithm

This section recalls briefly the optimization framework used in the companion paper [13] for designing low-complexity decimation filters $H(z)$ as a cascade of CP subfilters. We invite an interested reader to refer to [13] for further details, as well as for a list of key properties of CPs, and suggestions for the choice of low-complexity CPs. The main idea at the basis of the proposed design technique is to represent the frequency response $H(f_d)$ of the decimation filter $H(z)$ [see Fig. 1(b)] as the cascade $H(f_d) = \prod_{q=1}^{|S_{cp}|} C_q^{m_q}(f_d)$ of eligible CPs in the set S_{cp} ($|S_{cp}|$ is the cardinality of the set, i.e., the number of eligible CPs). The frequency response of a CP with index q is identified by $C_q(f_d)$, whereas m_q (with $m_q \geq 0\; \forall\, q$) is the integer order by which $C_q(f_d)$ appears in the cascade. The z-transfer functions of the CPs $C_q(z)$ with indexes $q \in \{61, \ldots, 200\}$ are shown in Table I.

The choice of eligible CPs in the set $q \in \{1, \ldots, 200\}$ is accomplished by searching for CPs featuring the two following properties: At least $p\%$ of zeros of $C_q(f_d)$ falls within folding bands defined in (1) [see Fig. 1(b)], and the polynomial $C_q(z)$ does not have zeros falling in a passband ranging from 0 to f_c.

Once the set S_{cp} of eligible CPs along with appropriate specifications (a passband ripple and folding band attenuations) has been identified, the design of the decimation filter $H(z)$ is based on the following optimization problem:

$$\min_{m_1,\ldots,m_{|S_{cp}|}} F\left(m_1, m_2, \ldots, m_{|S_{cp}|}\right) = \sum_{q=1}^{|S_{cp}|} m_q \cdot N_q$$

subject to :

$$\begin{aligned} &0) \quad \sum_{q=1}^{|S_{cp}|} m_q A_{d_q} \leq R_p \text{ (ripple)} \\ &1) \quad \sum_{q=1}^{|S_{cp}|} m_q A_s(1, q) \leq A_s \text{ (selectivity)} \\ &\ldots \quad \ldots \quad \ldots \\ &k_M) \quad \sum_{q=1}^{|S_{cp}|} m_q A_s(k_M, q) \leq A_s \end{aligned} \tag{4}$$

where R_p is the desired ripple [see (2)] in decibels, A_s is the target selectivity in folding bands [see (3)], N_q is the number of adders of the CP $C_q(z)$ (N_q is shown in Table I), $A_{d_q} = -\max_{f_d \in [0; f_c]}(|C_q(f_d)|_n)|_{dB}$ is the passband ripple of the CP $C_q(z)$, and $A_s(k, q) = \min_{f_d \in [(k/D)-f_c; (k/D)+f_c]}(|C_q(f_d)|_n)|_{dB}$ is the worst attenuation of the qth CP in S_{cp} within the kth folding band, with $k \in \{1, \ldots, k_M\}$ and k_M defined in (1) (the subscript n is used to signify that the polynomials are normalized to have a unity gain at a baseband). All these values (in decibels) are evaluated using frequency responses of the eligible CPs in S_{cp} and stored in look-up tables.

The solution of the mixed-integer linear programming optimization problem in (4) is the set of CP orders $\mathbf{m} = [m_1, \ldots, m_{|S_{cp}|}]^T$, minimizing the complexity cost function $F(m_1, m_2, \ldots, m_{|S_{cp}|})$ ($m_i = 0$ signifies the fact that the ith CP in S_{cp} is not employed in the synthesis of $H(z)$).

III. Design Examples, Implementation Issues, and Comparisons

This section presents some results obtained by applying the optimization algorithm in (4). For conciseness, in this brief, we will focus on the design of two decimation filters in Table II, although these considerations can be applied quite straightforwardly to the design of other decimation filters.

The results of the optimization problem in (4) are shown in Table II for various specifications of A_s and R_p. With reference to the two-stage architecture in Fig. 1(a), we solved the optimization problem by assuming that the residual decimation factor is ν (in other words, we assumed that $\rho = D \cdot \nu$).

Some details are in order. The first column in each row shows the decimation factor and the specifications A_s and R_p used in the design, whereas the second column presents the set of eligible CPs chosen according to the steps discussed in Section II and the optimal filter $H(z)$ designed with the optimization algorithm in (4). We notice in passing that $H(z)$ is always a linear-phase filter being a cascade of linear-phase constituent filters $C_p(z)$ even when recursive architectures are devised for each $C_p(z)$ (CPs are only apparently recursive, whereas in fact, they all only have zeros on the unit circle).

Before presenting the design examples, we point out that the design of multirate filters based on the extended set of CPs in Table I yields filters with a number of addition less or equal to the number obtained by the design based on only the first 104 CPs, as in [13]. As an instance, the filter $H(z) = C_{53}(z)C_{59}(z)C_{109}^2(z)$ corresponding to $D = 55$ and $\nu = 11$ in Table II requires only eight addition using the first 200 CPs, whereas it requires 11 addition when the design is accomplished on the first 104 CPs only.

TABLE I
CYCLOTOMIC POLYNOMIALS C_q FOR $q \in \{61, \ldots, 200\}$ ALONG WITH THE NUMBER OF ADDITIONS N_q

q	$C_q(z^{-1})$, N_q	q	$C_q(z^{-1})$, N_q	q	$C_q(z^{-1})$, N_q
61	$\frac{1-z^{-61}}{1-z^{-1}}$, 2	71	$\frac{1-z^{-71}}{1-z^{-1}}$, 2	81	$1+z^{-27}+z^{-54}$, 2
62	$\sum_{i=0}^{30}(-1)^i z^{-i} = \frac{1+z^{-31}}{1+z^{-1}}$, 2	72	$1-z^{-12}+z^{-24}$, 2	82	$\sum_{i=0}^{40}(-1)^i z^{-i} = \frac{1+z^{-41}}{1+z^{-1}}$, 2
63	$\frac{1+z^{-21}+z^{-42}}{1+z^{-3}+z^{-6}}$, 4	73	$\frac{1-z^{-73}}{1-z^{-1}}$, 2	83	$\frac{1-z^{-83}}{1-z^{-1}}$, 2
64	$1+z^{-32}$, 1	74	$\sum_{i=0}^{36}(-1)^i z^{-i} = \frac{1+z^{-37}}{1+z^{-1}}$, 2	84	$\frac{1-z^{-14}+z^{-28}}{1-z^{-2}+z^{-4}}$, 4
65	$\frac{1-z^{-1}-z^{-65}+z^{-66}}{1-z^{-5}-z^{-13}+z^{-18}}$, 6	75	$\frac{1-z^{-5}-z^{-75}+z^{-80}}{1-z^{-15}-z^{-25}+z^{-40}}$, 6	85	$\frac{1-z^{-1}-z^{-85}+z^{-86}}{1-z^{-5}-z^{-17}+z^{-22}}$, 6
66	$\frac{1-z^{-11}+z^{-22}}{1-z^{-1}+z^{-2}}$, 4	76	$\sum_{i=0}^{18}(-1)^i z^{-2i} = \frac{1+z^{-38}}{1+z^{-2}}$, 2	86	$\sum_{i=0}^{42}(-1)^i z^{-i} = \frac{1+z^{-43}}{1+z^{-1}}$, 2
67	$\frac{1-z^{-67}}{1-z^{-1}}$, 2	77	$\frac{1-z^{-1}-z^{-77}+z^{-78}}{1-z^{-7}-z^{-11}+z^{-18}}$, 6	87	$\frac{1-z^{-1}-z^{-87}+z^{-88}}{1-z^{-3}-z^{-29}+z^{-32}}$, 6
68	$\sum_{i=0}^{16}(-1)^i z^{-2i} = \frac{1+z^{-34}}{1+z^{-2}}$, 2	78	$\frac{1+z^{-1}+z^{-39}+z^{-40}}{1+z^{-3}+z^{-13}+z^{-16}}$, 6	88	$\sum_{i=0}^{10}(-1)^i z^{-4i} = \frac{1+z^{-44}}{1+z^{-4}}$, 2
69	$\frac{1-z^{-1}-z^{-69}+z^{-70}}{1-z^{-3}-z^{-23}+z^{-26}}$, 6	79	$\frac{1-z^{-79}}{1-z^{-1}}$, 2	89	$\frac{1-z^{-89}}{1-z^{-1}}$, 2
70	$\frac{1+z^{-1}+z^{-35}+z^{-36}}{1+z^{-5}+z^{-7}+z^{-12}}$, 6	80	$\frac{1+z^{-40}}{1+z^{-8}}$, 2	90	$\frac{1-z^{-15}+z^{-30}}{1-z^{-3}+z^{-6}}$, 4
91	$\frac{1-z^{-1}-z^{-91}+z^{-92}}{1-z^{-7}-z^{-13}+z^{-20}}$, 6	95	$\frac{1-z^{-1}-z^{-95}+z^{-96}}{1-z^{-5}-z^{-19}+z^{-24}}$, 6	99	$\frac{1+z^{-33}+z^{-66}}{1+z^{-3}+z^{-6}}$, 4
92	$\sum_{i=0}^{22}(-1)^i z^{-2i} = \frac{1+z^{-46}}{1+z^{-2}}$, 2	96	$1-z^{-16}+z^{-32}$, 2	100	$\sum_{i=0}^{4}(-1)^i z^{-10i} = \frac{1+z^{-50}}{1+z^{-10}}$, 2
93	$\frac{1-z^{-1}-z^{-93}+z^{-94}}{1-z^{-3}-z^{-31}+z^{-34}}$, 6	97	$\frac{1-z^{-97}}{1-z^{-1}}$, 2	101	$\frac{1-z^{-101}}{1-z^{-1}}$, 2
94	$\sum_{i=0}^{46}(-1)^i z^{-i} = \frac{1+z^{-47}}{1+z^{-1}}$, 2	98	$\sum_{i=0}^{6}(-1)^i z^{-7i} = \frac{1+z^{-49}}{1+z^{-7}}$, 2	102	$\frac{1-z^{-17}+z^{-34}}{1-z^{-1}+z^{-2}}$, 4
103	$\frac{1-z^{-103}}{1-z^{-1}}$, 2	104	$\sum_{i=0}^{12}(-1)^i z^{-4i} = \frac{1+z^{-52}}{1+z^{-4}}$, 2		
105	$\frac{1-z^{-7}+z^{-21}-z^{-28}+z^{-35}-z^{-49}+z^{-56}}{1-z^{-1}+z^{-3}-z^{-4}+z^{-5}-z^{-7}+z^{-8}}$, 12	115	$\frac{1-z^{-1}-z^{-115}+z^{-116}}{1-z^{-5}-z^{-23}+z^{-28}}$, 6	125	$(1-z^{-125})(1-z^{-25})^{-1}$, 2
106	$(1+z^{-53})(1+z^{-1})^{-1}$, 2	116	$(1+z^{-58})(1+z^{-2})^{-1}$, 2	126	$(1-z^{-21}+z^{-42})(1-z^{-3}+z^{-6})^{-1}$, 4
107	$(1-z^{-107})(1-z^{-1})^{-1}$, 2	117	$\frac{1+z^{-39}+z^{-78}}{1+z^{-3}+z^{-6}}$, 4	127	$\frac{1-z^{-127}}{1-z^{-1}}$, 2
108	$1-z^{-18}+z^{-36}$, 2	118	$(1+z^{-59})(1+z^{-1})^{-1}$, 2	128	$1+z^{-64}$, 1
109	$\frac{1-z^{-109}}{1-z^{-1}}$, 2	119	$\frac{1-z^{-1}-z^{-119}+z^{-120}}{1-z^{-7}-z^{-17}+z^{-24}}$, 6	129	$\frac{1+z^{-43}+z^{-86}}{1+z^{-1}+z^{-2}}$, 4
110	$\frac{1+z^{-1}+z^{-55}+z^{-56}}{1+z^{-5}+z^{-11}+z^{-16}}$, 6	120	$\frac{1-z^{-20}+z^{-40}}{1-z^{-4}+z^{-8}}$, 4	130	$\frac{1+z^{-1}+z^{-65}+z^{-66}}{1+z^{-5}+z^{-13}+z^{-18}}$, 6
111	$(1+z^{-37}+z^{-74})(1+z^{-1}+z^{-2})^{-1}$, 4	121	$(1-z^{-121})(1-z^{-11})^{-1}$, 2	131	$(1-z^{-131})(1-z^{-1})^{-1}$, 2
112	$(1+z^{-56})(1+z^{-8})^{-1}$, 2	122	$(1+z^{-61})(1+z^{-1})^{-1}$, 2	132	$(1-z^{-22}+z^{-44})(1-z^{-2}+z^{-4})^{-1}$, 4
113	$\frac{1-z^{-113}}{1-z^{-1}}$, 2	123	$\frac{1+z^{-41}+z^{-82}}{1+z^{-1}+z^{-2}}$, 4	133	$\frac{1-z^{-1}-z^{-133}+z^{-134}}{1-z^{-7}-z^{-19}+z^{-26}}$, 6
114	$(1-z^{-19}+z^{-38})(1-z^{-1}+z^{-2})^{-1}$, 4	124	$(1+z^{-62})(1+z^{-2})^{-1}$, 2	134	$(1+z^{-67})(1+z^{-1})^{-1}$, 2
135	$\frac{1+z^{-45}+z^{-90}}{1+z^{-9}+z^{-18}}$, 4	145	$\frac{1-z^{-1}-z^{-145}+z^{-146}}{1-z^{-5}-z^{-29}+z^{-34}}$, 6	155	$\frac{1-z^{-1}-z^{-155}+z^{-156}}{1-z^{-5}-z^{-31}+z^{-36}}$, 6
136	$(1+z^{-68})(1+z^{-4})^{-1}$, 2	146	$(1+z^{-73})(1+z^{-1})^{-1}$, 2	156	$(1-z^{-26}+z^{-52})(1-z^{-2}+z^{-4})^{-1}$, 4
137	$(1-z^{-137})(1-z^{-1})^{-1}$, 2	147	$\frac{1+z^{-49}+z^{-98}}{1+z^{-7}+z^{-14}}$, 4	157	$\frac{1-z^{-157}}{1-z^{-1}}$, 2
138	$(1-z^{-23}+z^{-46})(1-z^{-1}+z^{-2})^{-1}$, 4	148	$(1+z^{-74})(1+z^{-2})^{-1}$, 2	158	$(1+z^{-79})(1+z^{-1})^{-1}$, 2
139	$(1-z^{-139})(1-z^{-1})^{-1}$, 2	149	$(1-z^{-149})(1-z^{-1})^{-1}$, 2	159	$(1+z^{-53}+z^{-106})(1+z^{-1}+z^{-2})^{-1}$, 4
140	$\frac{1-z^{-14}+z^{-28}-z^{-42}+z^{-56}}{1-z^{-2}+z^{-4}-z^{-6}+z^{-8}} =$	150	$\frac{1-z^{-25}+z^{-50}}{1-z^{-5}+z^{-10}}$, 4	160	$\frac{1+z^{-80}}{1+z^{-16}}$, 2
	$= \frac{1+z^{-2}+z^{-70}+z^{-72}}{1-z^{-10}+z^{-14}+z^{-24}}$, 6				
141	$\frac{1+z^{-47}+z^{-94}}{1+z^{-1}+z^{-2}}$, 4	151	$\frac{1-z^{-151}}{1-z^{-1}}$, 2	161	$\frac{1-z^{-1}-z^{-161}+z^{-162}}{1-z^{-7}-z^{-23}+z^{-30}}$, 6
142	$\frac{1+z^{-71}}{1+z^{-1}}$, 2	152	$\frac{1+z^{-76}}{1+z^{-4}}$, 2	162	$1-z^{-27}+z^{-54}$, 2
143	$\frac{1-z^{-1}-z^{-143}+z^{-144}}{1-z^{-11}-z^{-13}+z^{-24}}$, 6	153	$\frac{1+z^{-51}+z^{-102}}{1+z^{-3}+z^{-6}}$, 4	163	$\frac{1-z^{-163}}{1-z^{-1}}$, 2
144	$1-z^{-24}+z^{-48}$, 2	154	$\frac{1+z^{-1}+z^{-77}+z^{-78}}{1+z^{-7}+z^{-11}+z^{-18}}$, 6	164	$\frac{1+z^{-82}}{1+z^{-2}}$, 2
165	$\frac{1-z^{-11}+z^{-33}-z^{-44}+z^{-55}-z^{-77}+z^{-88}}{1-z^{-1}+z^{-3}-z^{-4}+z^{-5}-z^{-7}+z^{-8}}$, 12	177	$\frac{1+z^{-59}+z^{-118}}{1+z^{-1}+z^{-2}}$, 4	189	$\frac{1+z^{-63}+z^{-126}}{1+z^{-9}+z^{-18}}$, 4
166	$(1+z^{-83})(1+z^{-1})^{-1}$, 2	178	$(1+z^{-89})(1+z^{-1})^{-1}$, 2	190	$\frac{1+z^{-1}+z^{-95}+z^{-96}}{1+z^{-5}+z^{-19}+z^{-24}}$, 6
167	$(1-z^{-167})(1-z^{-1})^{-1}$, 2	179	$(1-z^{-179})(1-z^{-1})^{-1}$, 2	191	$\frac{1-z^{-191}}{1-z^{-1}}$, 2
168	$(1-z^{-28}+z^{-56})(1-z^{-4}+z^{-8})^{-1}$, 4	180	$\frac{1-z^{-30}+z^{-60}}{1-z^{-6}+z^{-12}}$, 4	192	$1-z^{-32}+z^{-64}$, 2
169	$(1-z^{-169})(1-z^{-13})^{-1}$, 2	181	$(1-z^{-181})(1-z^{-1})^{-1}$, 2	193	$(1-z^{-193})(1-z^{-1})^{-1}$, 2
170	$\frac{1+z^{-1}+z^{-85}+z^{-86}}{1+z^{-5}+z^{-17}+z^{-22}}$, 6	182	$\frac{1+z^{-1}+z^{-91}+z^{-92}}{1+z^{-7}+z^{-13}+z^{-20}}$, 6	194	$\frac{1+z^{-97}}{1+z^{-1}}$, 2
171	$\frac{1+z^{-57}+z^{-114}}{1+z^{-3}+z^{-6}}$, 4	183	$\frac{1+z^{-61}+z^{-122}}{1+z^{-1}+z^{-2}}$, 4	195	$\frac{1-z^{-13}+z^{-39}-z^{-52}+z^{-65}-z^{-91}+z^{-104}}{1-z^{-1}+z^{-3}-z^{-4}+z^{-5}-z^{-7}+z^{-8}}$, 12
172	$\frac{1+z^{-86}}{1+z^{-2}}$, 2	184	$\frac{1+z^{-92}}{1+z^{-4}}$, 2	196	$\frac{1+z^{-98}}{1+z^{-14}}$, 2
173	$\frac{1-z^{-173}}{1-z^{-1}}$, 2	185	$\frac{1-z^{-1}-z^{-185}+z^{-186}}{1-z^{-5}-z^{-37}+z^{-42}}$, 6	197	$\frac{1-z^{-197}}{1-z^{-1}}$, 2
174	$\frac{1-z^{-29}+z^{-58}}{1-z^{-1}+z^{-2}}$, 4	186	$\frac{1-z^{-31}+z^{-62}}{1-z^{-1}+z^{-2}}$, 4	198	$\frac{1-z^{-33}+z^{-66}}{1-z^{-3}+z^{-6}}$, 4
175	$\frac{1-z^{-5}-z^{-175}+z^{-180}}{1-z^{-25}-z^{-35}+z^{-60}}$, 6	187	$\frac{1-z^{-1}-z^{-187}+z^{-188}}{1-z^{-11}-z^{-17}+z^{-28}}$, 6	199	$\frac{1-z^{-199}}{1-z^{-1}}$, 2
176	$\frac{1+z^{-88}}{1+z^{-8}}$, 2	188	$\frac{1+z^{-94}}{1+z^{-2}}$, 2	200	$\frac{1+z^{-100}}{1+z^{-20}}$, 2

As a first example, consider the following filter:

$$H(z) = C_{25}(z)C_{31}^2(z) = \frac{1-z^{-25}}{1-z^{-5}} \cdot \left(\frac{1-z^{-31}}{1-z^{-1}}\right)^2 \qquad (5)$$

in Table II, attaining the specifications $A_s = 40$ dB and $R_p = 1$ dB for $D = 25$. Its architecture in a multistage decimation structure is shown in Fig. 2(a). From the commutative property employed in [5], the polynomial $1 - z^{-25}$ can be moved across

TABLE II
OPTIMIZATION RESULTS

$D = 25, \nu = 4$ $p = 20\%$	Set of eligible CPs: 5, 8, 9, 11, 12, 14, 16, 17, 18, 19, 21, 22, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 66, 67, 68, 69, 70, 71, 72, 73, 74, 76, 77, 78, 79, 80, 81, 82, 83, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 116, 117, 118, 119, 120, 121, 122, 123, 124, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 176, 177, 178, 179, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199
$A_s = 40,\ R_p = 1$dB	$H(z) = C_{25}(z)C_{31}^2(z)$
$D = 55, \nu = 11$ $p = 5\%$	Set of eligible CPs: 2, 5, 11, 23, 24, 26, 27, 28, 29, 31, 32, 34, 36, 37, 38, 39, 41, 42, 46, 47, 48, 49, 51, 52, 54, 55, 56, 57, 58, 61, 62, 63, 64, 67, 68, 69, 71, 72, 73, 74, 76, 78, 79, 81, 82, 83, 84, 86, 87, 89, 91, 92, 93, 94, 96, 97, 98, 101, 102, 103 106, 107, 109, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 199, 200
$A_s = 90,\ R_p = 0.3$dB	$H(z) = C_{53}(z)C_{59}(z)C_{109}^2(z)$
$D = 61, \nu = 11$ $p = 5\%$	Set of eligible CPs: 2, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 64, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 121, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200
$A_s = 100,\ R_p = 0.4$dB	$H(z) = C_{61}^3(z)C_{127}(z)$
$D = 64, \nu = 4$ $p = 20\%$	Set of eligible CPs: 2, 4, 8, 9, 11, 15, 16, 17, 18, 19, 21, 22, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 41, 42, 43, 44, 45, 47, 49, 50, 51, 53, 54, 55, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 93, 94, 95, 97, 98, 99, 100, 101, 102, 103, 105, 106, 107, 108, 109, 110, 111, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 185, 186, 187, 188, 189, 190, 191, 193, 194, 195, 196, 197, 198, 199, 200
$A_s = 60,\ R_p = 1$dB	$H(z) = C_{61}(z)C_{64}(z)C_{67}^2(z)$
$D = 64, \nu = 8$ $p = 5\%$	Set of eligible CPs: 2, 4, 8, 16, 17, 19, 21, 23, 25, 27, 29, 31, 32, 33, 34, 35, 37, 38, 39, 41, 42, 43, 45, 46, 47, 49, 50, 51, 53, 54, 55, 57, 58, 59, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 73, 74, 75, 76, 77, 78, 79, 81, 82, 83, 84, 85, 86, 87, 89, 90, 91, 92, 93, 94, 95, 97, 98, 99, 100, 101, 102, 103, 105, 106, 107, 108, 109, 110, 111, 113, 114, 115, 116, 117, 118, 119, 121, 122, 123, 124, 125, 126, 127, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 193, 194, 195, 196, 197, 198, 199, 200
$A_s = 80,\ R_p = 0.5$dB	$H(z) = C_{67}^3(z)C_{131}(z)$
$D = 65, \nu = 10$ $p = 5\%$	Set of eligible CPs: 2, 5, 13, 21, 22, 23, 24, 27, 28, 29, 31, 32, 33, 34, 36, 37, 38, 41, 42, 43, 44, 46, 47, 48, 49, 51, 53, 54, 56, 57, 58, 59, 61, 62, 63, 64, 65, 66, 67, 68, 71, 72, 73, 74, 76, 77, 79, 81, 82, 83, 84, 86, 87, 88, 89, 92, 93, 94, 96, 97, 98, 99, 100, 101, 102, 103, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 183, 184, 186, 187, 188, 189, 190, 191, 192, 193, 194, 196, 197, 198, 199, 200
$A_s = 100,\ R_p = 0.4$dB	$H(z) = C_{64}(z)C_{67}^2(z)C_{109}(z)C_{131}(z)$

the decimator by 25 obtaining a more efficient architecture shown in Fig. 2(b). Moreover, upon splitting the decimator in two cascaded decimators, each decimating by 5, the cell $1/(1-z^{-5})$ can be moved on the right side of the first decimator by 5, as depicted in Fig. 2(c). This is the most efficient architecture for this filter since the original subfilter $1-z^{-25}$ is now a differentiator $1-z^{-1}$ operating at a sampling rate 25 times lower than that in the architecture Fig. 2(a), whereas the subfilter $1/(1-z^{-5})$ becomes an integrator $1/(1-z^{-1})$ operating at a sampling rate five times lower than that in the architecture Fig. 2(a).

The second example concerns the implementation of the filter $H(z)$ as follows:

$$C_{61}(z)C_{64}(z)C_{67}^2(z) = \frac{1-z^{-61}}{1-z^{-1}}(1+z^{-32})\left(\frac{1-z^{-67}}{1-z^{-1}}\right)^2 \quad (6)$$

in Table II, attaining the specifications $A_s = 60$ dB and $R_p = 1$ dB for $D = 64$ and $\nu = 4$. Its architecture in a multistage

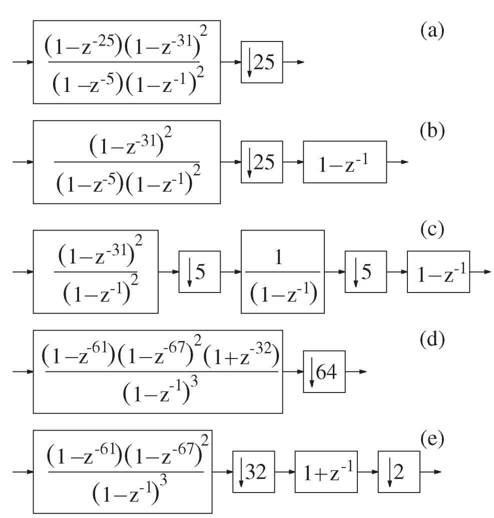

Fig. 2. (a)–(c) Efficient architectures for implementing the decimation stage embedding $H(z)$ for $D = 25$. (d) and (e) Efficient architectures for implementing the decimation stage embedding $H(z)$ for $D = 64$.

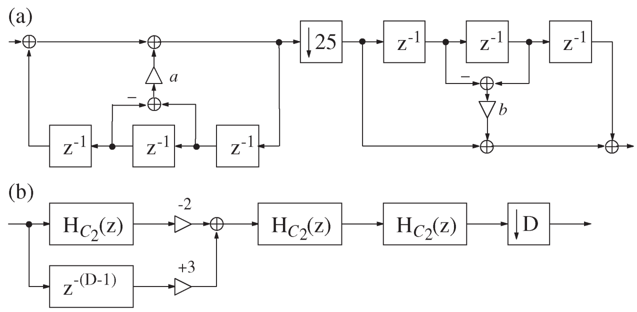

Fig. 3. (a) Architecture of the third-order MSDF filter ($a = 1 + 2\cos(\alpha)$, $b = 1 + 2\cos(D\alpha)$, $\alpha = (\pi q/\rho)$, $q = 0.79$). (b) Architecture of the SCD filters.

decimation structure is shown in Fig. 2(d). Upon splitting the decimator in two cascaded decimators, as shown in Fig. 2(e), the cell $(1 + z^{-32})$ can be moved to the right side of the first decimator by 32, obtaining the architecture shown in Fig. 2(e). This is the most efficient architecture for this filter since the original subfilter $1 + z^{-32}$ becomes an adder $(1 + z^{-1})$ operating at a sampling rate 32 times lower than that in the architecture in Fig. 2(d).

In the remainder of this section, we discuss some comparisons with other two techniques proposed recently in the literature.

Consider the modified-sinc decimation filters (MSDF) proposed in [14] along with the specifications $A_s = 40$ dB and $R_p = 1$ dB noted in Table II for the case $D = 25$. In order to attain the underlined specifications when $D = 25$, a third-order MSDF is required. The MSDF filter, which can be implemented using the recursive architecture proposed in [14] (with the optimal value $q = 0.79$) and shown in Fig. 3(a), presents a real multiplier a operating at high frequency as well as a real multiplier b operating at reduced frequency after decimation by 25. A comparison between the architecture shown in Fig. 3(a) and the one related to the designed filter $H(z)$ and noted in Fig. 2(c) shows that the latter architecture for implementing the proposed filter $H(z)$ is much more computationally efficient than the equivalent MSDF, in that it does not require any real multiplication.

Let us focus on the sharpened cascaded integrator–comb decimation (SCD) filters proposed in [15]. An Lth-order SCD filter decimating by D has the following frequency response:

$$\left|H(e^{j\omega})\right| = \left| 3\left(\frac{\sin(\pi f_d D)}{D\,\sin(\pi f_d)}\right)^{2L} - 2\left(\frac{\sin(\pi f_d D)}{D\,\sin(\pi f_d)}\right)^{3L} \right|. \tag{7}$$

In order to meet the specifications $A_s = 40$ dB and $R_p = 1$ dB, an SCD filter with $L = 2$ is needed. The architecture implementing such filter is depicted in Fig. 3(b), whereby $H_{C_2}(z)$ is a second-order comb filter with z-transfer function as

$$H_{C_2}(z) = \frac{1}{D^2}\left(\frac{1 - z^{-D}}{1 - z^{-1}}\right)^2$$

where $D = 25$ in this specific setup. Upon identifying with f_o the sampling rate at the input, the computational complexity required by the filter in Fig. 3(b) is eight adders operating at f_o (two of them required to implement the integer multipliers $\times\ 2$ and $\times\ 3$) and six adders operating at the rate $f_o/25$. On the other hand, the proposed filter $H(z)$ in Fig. 2(c) requires four adders at a high rate f_o, one at a rate $f_o/5$, and one adder at $f_o/25$, making it more effective compared with the SCD architecture.

IV. Conclusion

This brief has proposed the design of multiplierless decimation filters based on an extended search of CPs. The aim was twofold. On one hand, it has proposed an optimization framework for the design of constituent decimation filters in multistage decimation architectures using as basic building blocks all the CPs with indexes extended to the set $\{1, \ldots, 200\}$. On the other hand, this brief has provided a number of useful techniques for designing optimized filters in a variety of architectures. Design guidelines were provided with the aim to reduce still further the computational complexity of the decimation filters.

References

[1] M. Laddomada, F. Daneshgaran, M. Mondin, and R. M. Hickling, "A PC-based software receiver using a novel front-end technology," *IEEE Commun. Mag.*, vol. 39, no. 8, pp. 136–145, Aug. 2001.
[2] F. Daneshgaran and M. Laddomada, "Transceiver front-end technology for software radio implementation of wideband satellite communication systems," *Wireless Pers. Commun.*, vol. 24, no. 12, pp. 99–121, Dec. 2002.
[3] S. R. Norsworthy, R. Schreier, and G. C. Temes, *Delta–Sigma Data Converters, Theory, Design, and Simulation*. Piscataway, NJ: IEEE Press, 1997.
[4] R. E. Crochiere and L. R. Rabiner, "Interpolation and decimation of digital signals—A tutorial review," *Proc. IEEE*, vol. 69, no. 3, pp. 300–331, Mar. 1981.
[5] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial," *Proc. IEEE*, vol. 78, no. 1, pp. 56–93, Jan. 1990.
[6] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[7] R. J. Hartnett and G. F. Boudreaux-Bartels, "On the use of cyclotomic polynomial prefilters for efficient FIR filter design," *IEEE Trans. Signal Process.*, vol. 41, no. 5, pp. 1766–1779, May 1993.
[8] H. J. Oh and Y. H. Lee, "Design of efficient FIR filters with cyclotomic polynomial prefilters using mixed integer linear programming," *IEEE Signal Process. Lett.*, vol. 3, no. 8, pp. 239–241, Aug. 1996.
[9] H. J. Oh and Y. H. Lee, "Design of discrete coefficient FIR and IIR digital filters with prefilter-equalizer structure using linear programming," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 6, pp. 562–565, Jun. 2000.
[10] K. Supramaniam and Y. Lian, "Complexity reduction for frequency-response masking filters using cyclotomic polynomial prefilters," in *Proc. IEEE ISCAS*, May 21–24, 2006, pp. 3297–3300.
[11] Y. Voronenko and M. Püschel, "Multiplierless multiple constant multiplication," *ACM Trans. Algorithms*, vol. 3, no. 2, pp. 1–38, May 2007.
[12] L. Aksoy, E. da Costa, P. Flores, and J. Monteiro, "Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 6, pp. 1013–1026, Jun. 2008.
[13] M. Laddomada, "Design of multistage decimation filters using cyclotomic polynomials: Optimization and design issues," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 7, pp. 1977–1987, Aug. 2008.
[14] L. Lo Presti, "Efficient modified-sinc filters for sigma–delta A/D converters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 11, pp. 1204–1213, Nov. 2000.
[15] A. Y. Kwentus, Z. Jiang, and A. N. Willson, Jr., "Application of filter sharpening to cascaded integrator–comb decimation filters," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 457–467, Feb. 1997.